

Dispečerský modul pro dopravní monitorovací systém

Dispatching Module for Traffic Monitoring System

Zadání bakalářské práce

Student:

Ondřej Vybíral

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

Dispečerský modul pro dopravní monitorovací systém
Dispatching Module for Traffic Monitoring System

Zásady pro vypracování:

Cílem bakalářské práce je vytvoření dispečerského modulu pro dopravní monitorovací systém vyvíjený společností GX Solutions ve spolupráci s VŠB-TU Ostrava. Tento modul bude určen pro online sledování vozidel, a to i s možností zobrazení historie. Modul bude využívat externí informace, jakými jsou POI, dopravní informace, počasí, kamery atd. Při vytváření bakalářské práce student obdrží základní knihovny pro komunikaci s databází a dalšími existujícími částmi systému.

Jednotlivé body bakalářské práce:

1. Analýza současného stavu v řešené problematice a porovnání stávajícího řešení firmy GX Solutions s dalšími existujícím softwary.
2. Implementace databázové vrstvy pro dispečerský modul včetně potřebných funkcí pro import a správu dat.
3. Vytvoření uživatelského rozhraní dispečerského modulu.
4. Testování vytvořené aplikace s využitím reálných dat.

Seznam doporučené odborné literatury:

- [1] GX Solutions. Manuál na obsluhu SW Truck Data Memory 2010 v2.2
- [2] Jednotný systém dopravních informací pro ČR (JSDI): <http://www.dopravniinfo.cz/jsdi>

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Kateřina Slaninová, Ph.D.**

Datum zadání: 01.09.2014

Datum odevzdání: 07.05.2015



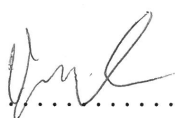
doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 20. července 2015


.....

Rád bych poděkoval Ing. Kateřině Slaninové, Ph.D. za vedení bakalářské práce. Dále bych chtěl poděkovat panu Ing. Janu Martinoviči, Ph.D. za možnost účastnit se na tomhle projektu. V poslední řadě bych chtěl poděkovat rodině a nejbližším přátelům za podporu.

Abstrakt

Bakalářská práce se zabývá vytvořením dispečerského modulu pro dopravní monitorovací systém vyvíjený společností GX Solutions ve spolupráci s VŠB-TU Ostrava. Tento modul bude určen pro online sledování vozidel, a to i s možností zobrazení historie. Modul bude využívat externí informace, jakými jsou POI, dopravní informace, počasí, kamery atd.

Klíčová slova: bakalářská práce, dispečerský modul, GX Solutions, devexpress, dopravní informace, logistika, systém pro sledování dopravy, zařízení, vozidla, historie pozic, online pozice

Abstract

This bachelor's thesis deals with the creation of dispatching module for traffic monitoring system developed by GX Solutions company in collaboration with the VŠB-TU Ostrava. This module is designed for online tracking of vehicles, also with the possibility of view of history. The module will utilize external information, such as POI, traffic information, weather, cameras etc.

Keywords: bachelor's thesis, dispatching module, traffic information, logistics, devices, vehicles, traffic monitoring system, history positions, online positions, devexpress, GX Solutions

Seznam použitých zkratek a symbolů

JSDI	–	Jednotný systém dopravních informací
FCD	–	Floating Car Data
API	–	Application Programming Interface
WCF	–	Windows Communication Foundation
BCD	–	Berkeley Software Distribution
ACID	–	Atomicity, Consistency, Isolation, Durability

Obsah

1	Úvod	5
2	Analýza konkurence	6
2.1	GPS Dozor, AutoGPS	6
2.2	WebDispečink, KomTes	7
2.3	Fleetware Web	8
2.4	ONI System	9
2.5	LogisCare	10
2.6	GX Solutions	11
2.7	Zhodnocení	12
3	Architektura projektu	14
4	Implementace databázové vrstvy	15
4.1	Použité technologie	15
4.2	Uživatelské nastavení	15
4.3	Online pozice vozidel	16
4.4	Historie pozic zařízení	19
5	Vytvoření uživatelského rozhraní dispečerského modulu	21
5.1	Návrh	22
5.2	Implementace	24
6	Testování vytvořené aplikace	32
7	Závěr	33
8	Reference	34
	Přílohy	34
A	Příloha č. 1 - Ukázka konkurenčních systémů	35

Seznam tabulek

1	Informace poskytované dispečerovi	12
2	Zobrazení historie vozidla	13
3	Uživatelské nastavení	13

Seznam obrázků

1	GPS Dozor - Popisek ikonky I	6
2	GPS Dozor - Popisek ikonky II	6
3	WebDispečink, Komtes - Informace o vozidle	7
4	Fleetware - Panel informace	9
5	ONI System - Panel se seznamem vozidel	10
6	LogisCare - Podrobnosti o vozidle	10
7	GX Solutions - Detailní informace z jednotky	11
8	Diagram komponent	14
9	Koncepтуální model	16
10	Dispečerský modul - Diagram případů užití	21
11	Wireframe - Dispečerský modul	22
12	Wireframe - Mapové nastavení	23
13	Model-View-Controller	24
14	Náhled generované ikonky	27
15	Mapa - Ovládací panel	30
16	Mapa - Mapový výřez	31
17	GPS Dozor - Náhled dispečerského modulu	35
18	Komtest - Náhled dispečerského modulu	35
19	Komtest - Průběh dne	36
20	Fleetware - Náhled aplikace	36
21	ONI System - Dispečerský modul	37
22	LogisCare - Náhled aplikace	37
23	GX Solutions - Náhled aplikace	38

Seznam výpisů zdrojového kódu

1	Online pozice - Zjištění přiřazených sloupců	17
2	Online pozice - Zjištění nastavení pro ikonky	17
3	Online pozice - Získání dat pro vybraná vozidla	17
4	Online pozice - Získání dat podle sloupců	18
5	Historie pozic - Zjištění nastavení pro ikonky	19
6	Historie pozic - Získání dat pro vozidlo a jednotky	20
7	Historie pozic - Získání pozic	20
8	Controller - PartialView pro tabulku pozic	26
9	Náhled generovaného SVG	27
10	Vkládání Overlay	28
11	Křivka historie	29
12	Mapa - Získání aktuálního výřezu	30

1 Úvod

Cílem bakalářské práce je vytvoření dispečerského modulu pro dopravní monitorovací systém vyvíjený společností GX Solutions ve spolupráci s VŠB-TU Ostrava. Monitorovací software se věnuje monitorováním vozidel a strojů. Nabízí také funkce jako plánování přeprav, navigaci vozidel, přehledy, hlídač událostí nad vozidly a jiné. Kvůli komplexnosti systému je vývoj rozdělen na moduly.

Systém je vyvíjen v týmové spolupráci, kdy každý člen vyvíjí jeden modul. Některé moduly mají mezi sebou závislost, část funkcionality jednoho modulu závisí na funkcionalitě druhého modulu.

Mým cílem je vytvoření dispečerského modulu. Dispečerský modul je určen pro on-line sledování pozic vozidel a zobrazování aktuálních informací z vozidel, dále umožňuje zobrazení historií pozic vozidel a zobrazení externích informací jako je počasí, JSDI [1], FCD a kamerové informace z křižovatek. Ve spolupráci s jinými moduly například zobrazuje historii jízd vozidel, zobrazování vytvářené přepravy, umožňuje vyhledávání tras a navigaci.

K vytvoření dispečerského modulu je potřeba několika kroků. Pro lepší pochopení dispečerského modulu a jeho funkcionality proběhla analýza konkurence, která srovnává konkurenční systémy se stávajícím produktem firmy GX Solutions. Analýze je věnována Kapitola 2.

V kapitole 3. je popsána architektura systému. Kapitola 4 se věnuje implementaci databázové vrstvy, která řeší načítání dat pro dispečerský modul z databáze. Kapitola 5 se věnuje vytvoření uživatelského rozhraní, což zahrnuje návrh a poté implementaci funkčnosti. Poslední Kapitola 6 je věnovaná testování vytvořeného modulu.

2 Analýza konkurence

Stav motoru zobrazuje blikající zelená ikonka, která se překrývá s textem. Typ jízdy je zobrazen po najetí kurzoru na ikonku. Modul dále obsahuje graf, který zobrazuje rychlost během časového období, které je nastaveno u daného vozidla, a dále také jednoduché nastavení mapy.










Nevýhody této aplikace spočívají v téměř žádné manipulaci se seznamem vozidel, kromě jednoduchého vyhledávání. Dále aplikace obsahuje Street view pohled, který je prakticky nevyužitelný. Zobrazuje se s vybraným vozidlem. Trasování jízdy by nemuselo být časově omezené na minimum zobrazených informací z vozidla.

Menší výhodou může být ikonka vozidla na mapě, která není nesmyslně velká jako u jiných aplikací.

2.2 WebDispečink, KomTes

Aplikace se zabývá spíše souhrny řidičů, vozidel, statistikami a nejrůznějšími analýzami. Dispečerský modul byl pro účely analýzy rozdělen do čtyř částí: na část věnující se výběru vozidel a seskupování, část o informacích z vozidla, část věnující se přehledům a na samotnou mapu s vozidly, viz Obrázek č.18 v Příloze č. 1.

Na mapě jsou zobrazena vybraná vozidla jako ikonky aut, které jsou u některých až zbytečně velké a při oddálení mapy zabírají zbytečně moc místa. Ikonka také blikáním indikuje pohyb vozidla. Trasování aktuální jízdy je řešeno barevnou křivkou a je časově omezené, nezobrazuje se celá jízda. Při trasování více vozidel jsou křivky barevně rozlišené. U ikonky vozidla je zobrazen popis s informacemi o vozidle. V nastavení lze množství těchto informací měnit, také samotná ikonka jde změnit, k dispozici jsou ale předdefinované ikony. První část působí dobře, dispečer má možnost si vozidla seskupit a zobrazovat podle potřeb. Druhá část, týkající se informací o vozidle, je také dobře řešena. Dispečer vidí informace stále na jednom místě, viz Obrázek č.3.

Informace o vozidle	
 RZ:	2M50014_1
	Volkswagen AG
	Olomouc
	- Olomouc podskupina
 Řidič:	Řidič AETR
 Stav:	Odpočinek
 Navigace:	odpojeno
 Místo:	CZ Hněvotín
 Ulice:	570
 km :	28340,70
 PHM :	50,00 l
 Rychlost:	72 km/h
Čas pozice:	22.09.2014 17:35:44
Čas výjezdu:	22.09.2014 17:23:42 (12)
Zem. šířka:	49° 34' 05,768" N
Zem. délka:	17° 11' 39,368" E
výška:	258 m
Dveře zadní:	Zavřeno
Dveře boční:	Zavřeno
Práce:	Pracuje

Obrázek 3: WebDispečink, Komtes - Informace o vozidle

Třetí část obsahuje vždy pro vybrané vozidlo knihu jízd, průběh dne, náklady, body dosažení, došlé zprávy a vlastní místa. Kniha jízd obsahuje klasický přehled jízd pro zadané období s možnostmi nechat si jízdu vypsat nebo zobrazit. Zajímavější je záložka Průběh dne, která obsahuje graf, kde jsou informace o stavu nádrže, stavu motoru a rychlosti během celého dne, viz Obrázek č.19 v Příloze č. 1.

Záložka Náklady zobrazuje veškeré náklady vybraného vozidla, převážně tankování. Záložka Body dosažení zobrazuje seznam bodů, které nejsou v demo verzi přístupné. Záložka Došlé zprávy z jednotky vozidla obsahuje většinou jen vypnutí/nastartování motoru. Záložka Vlastní místa obsahuje jen seznam všech míst, které lze samozřejmě zobrazit na mapě.

Nevýhoda této aplikace spočívá především ve špatném zobrazování vozidel na mapě. Ikonky jsou zbytečně velké, popisové okno nejde příliš měnit. Trasování aktuální jízdy by mohlo zobrazovat celou jízdu.

Výhoda aplikace spočívá především ve správě aut, kde je možné auta seskupovat do složek a minimalizovat tak seznam. Také se jako zajímavý jeví pevně daný modul s informacemi o vozidle. Celá třetí část je dle našeho názoru v dispečerském modulu celkem zbytečná. Lze ji však téměř celou skrýt.

2.3 Fleetware Web

Jedná se o aplikaci věnující se především monitoringu vozidel, v rozšířené verzi Fleetware 6+ nabízí modul zakázek ke sledování a vyhodnocování zakázek, plánování jízd a další. Dispečerský modul je rozdělen do několika panelů. Panel s mapou a polohou vozidel, panel ze seznamem všech vozidel, panel s informacemi vybraného vozidla, panel s nastavením mapy a uživatelských bodů, viz Obrázek č.20 v Příloze č. 1.

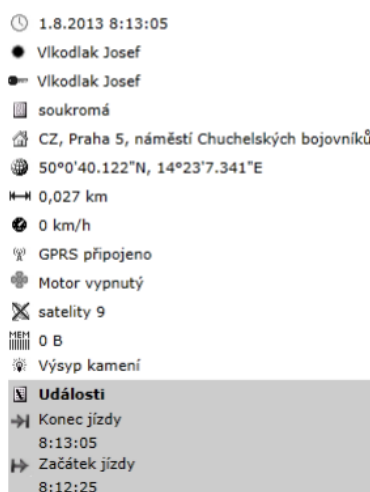
Mapový podklad by měl být od společnosti Radium. Vzhled mapy není příliš uživatelsky přívětivý, jsou na ní zbytečně velké popisky měst, části měst, barevné zvýraznění dálnic a rychlostních silnic. Celkově mapa působí dojmem, že se k tomuto účelu vůbec nehodí. Je zde sice možnost přepnout mapu na černobílý podklad, ale velikost zůstává stále stejná.

Zobrazení polohy vozidla je řešeno ikonkou šipky, která indikuje pohyb vozidla i směr. K ikonce lze v nastavení zapnout zobrazení popisků. Trasování aktuální jízdy je řešeno vyznačením křivky na mapě, která je časově omezená a nezobrazuje se tak celá jízda. Barva trasování je stejná pro více vozidel.

Panel se seznamem všech vozidel obsahuje vozidla a několik informací o nich (stav motoru, rychlost, účel, řidič, lokalita, poslední akt., staří údajů), panel obsahuje možnost filtrování vozidel podle nejrůznějších kritérií.

Panel informace o vozidle obsahuje přiblížený pohled mapy na vozidlo a detailnější informace z vozidla, viz Obrázek 4.

Panel s nastavením mapy obsahuje jednoduché nastavení popisků u ikoněk a trasování vozidla. Dále obsahuje záložku Uživatelské oblasti, kde lze oblasti spravovat a přidávat, bohužel v demo verzi je tato funkcionality omezená.



Obrázek 4: Fleetware - Panel informace

Nevýhodu aplikace vidíme ve špatném mapovém podkladu a trasování jízdy. V panelu s informacemi se nedala zmenšit nebo vypnout mapka s přiblížením na vozidlo. Přiblížení může být v některých případech užitečné, pokud máme hlavní mapu oddálenou na více vozidel. Panel se seznamem vozidel také není řešen příliš efektivně. Jsou v něm obsaženy informace, které se nachází hned ve vedlejším panelu. Také výběr vozidel pro jejich současné zobrazování mohlo být lépe řešeno než přes klávesu Ctrl.

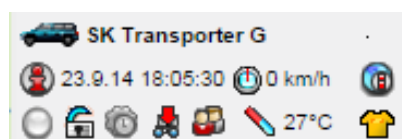
Výhody bychom hledali dlouho. Jediné co lze zmínit je, že v této aplikaci je možné částečně upravovat vzhled aplikace, jednotlivé panely jdou přepnout do záložek a nezabírají místo.

2.4 ONI System

Systém je v základní verzi zaměřen především na monitoring vozidel, ale obsahuje také jednoduchou komunikaci s řidičem. V rozšířených verzích obsahuje především bezpečnostní řešení. Dispečerský modul je velice jednoduchý, obsahuje samotnou mapu s polohou vozidel a postranní panel se seznamem vozidel a indikací stavu. Systém nabízí mapové podklady od společnosti Google a Plan Studio (viz Obrázek č.21 v Příloze č. 1). Na mapě jsou zobrazena vozidla pomocí jednoduché ikonky aut s názvem vozidla. Na mapě jsou vždy zobrazena všechna vozidla, jejich počet nelze nijak měnit. Pokud se vozidlo pohybuje zobrazí se uprostřed ikonky šipka. Systém nevykresluje aktuální trasu vozidla. Panel se seznamem vozidel zobrazuje všechna vozidla v bloku se třemi řádky, viz Obrázek 5.

První řádek obsahuje název vozidla, druhý řádek umožňuje přepnout pohled mapy na vozidlo, čas poslední polohy, aktuální rychlost a ONI dotaz. Třetí řádek indikuje poplach, zamknutí, univerzální vstup, stav motoru, typ jízdy, teplotu, sítěžení.

Dispečer zde má k dispozici jednoduchou komunikaci s řidičem prostřednictvím zasílání textových zpráv.



Obrázek 5: ONI System - Panel se seznamem vozidel

Nevýhodu systému vidíme především v chybějícím trasování jízdy. Také ikonky vozidel by mohly být lépe řešené, především jejich neustálé blikání, které nic neindikuje. Informace o stavu vozidla by mohly být také lépe řešeny než ikonkami. Práce se seznamem vozidel tu téměř žádná není, kromě jednoduchého vyhledávání.

Výhoda je snad jen v komunikaci s řidičem pomocí zpráv.

2.5 LogisCare

Aplikace věnující se monitoringu vozidel s jednoduchou komunikací s řidičem a napojením na plánovač tras od společnosti TimoCom.

Mapový podklad by měl být od OpenLayers, polohu vozidla zobrazuje ikonka auta s názvem vozidla. Trasování aktuální jízdy barevně rozlišuje rychlost jízdy a jednoduchými šipkami ukazuje směr jízdy, trasování zobrazuje aktuální jízdu od nastartování motoru. Pokud je vozidlo v pohybu, indikuje tento stav ikonka, u které se rozblíká modrý rámeček. Na mapě jsou vždy zobrazena všechna vozidla ze seznamu, viz Obrázek č.22 v Příloze č. 1. Levý panel obsahuje jednoduché vyhledávání a filtrování mezi vozidly, nachází se zde i jednoduchý modul pro komunikaci s řidičem. Ten se zobrazí pouze v případě, že má vozidlo nainstalovanou jednotku od společnosti Garmin. Spodní panel obsahuje seznam všech nebo filtrovaných vozidel se základními informacemi jako stav, typ jízdy, název, SPZ, čas poslední polohy, staří posledního stavu, typ zprávy, lokace. Pravý panel obsahuje detailnější informace z vybraného vozidla, viz Obrázek 6:



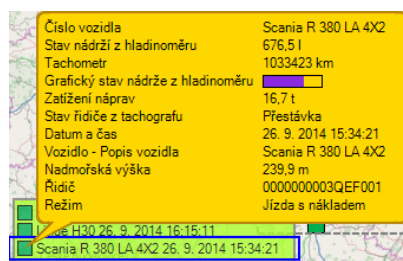
Obrázek 6: LogisCare - Podrobnosti o vozidle

Výhoda může být v trasování jízd, kdy se zobrazuje celá jízda a šipky zobrazují sběr dat v daném bodě, čas a rychlost. Komunikace s řidičem pomocí GPRS může být také výhodou. Ikonky vozidel by mohly být řešeny lépe. Mohla by být k dispozici resize panelů.

2.6 GX Solutions

Dispečerský modul je rozdělen na dvě části, levá část s filtrem a výběrem jednotek a pravá část, kam se zobrazují veškeré funkce, nabídky z menu, nastavení a především online sledování. Záložka online sledování obsahuje seznam jednotek a k nim informace (viz Obrázek č.23 v Příloze č. 1), plánovač tras a samotnou mapu s jednotkami.

Jednotky se na mapě zobrazují jako dynamické ikonky, které mění svůj tvar a barvu podle stavu. K ikonkám se zobrazují popisky s názvem a datem poslední aktualizace. Velice efektivní je seskupování popisků pod sebe, pokud je více jednotek poblíž sebe. Po najetí kurzoru na popisek se zobrazí detailní informace z jednotky, viz Obrázek 7.



Obrázek 7: GX Solutions - Detailní informace z jednotky

Mapa se dá přepnout na satelitní, terénní, nebo standardní. Je tu možnost uložit si aktuální výřez mapy a po té rychleji přepínat pohledy.

Levá část modulu filtruje jednotky podle typu a označené jednotky se zobrazují na mapě, po kliknutí na jednu z označených jednotek se pohled mapy přenese na jednotku. K označeným jednotkám je tu několik možností od zobrazení knihy jízd, sestavy, historie pozic, které se ještě ovládají časovým filtrem ve spodní části.

Výhoda aplikace spočívá především v ikonkách jednotek, ze kterých jde ihned vidět základní stav jednotky. Seskupování popisků je také velice efektivní, informační panel po najetí kurzoru není přehnaně velký. Výhodou také je, že u veškerých modulů lze změnit velikost a přizpůsobit si tak alespoň trochu celý dispečerský modul.

2.7 Zhodnocení

Zhodnocení všech analyzovaných firem formou tabulek. Tabulka č. 1 srovnává poskytování informací dispečerovi, Tabulka č. 2 se věnuje zobrazení historie pozic vozidel a Tabulka č. 3 uživatelskému nastavení.

2.7.1 Informace poskytované dispečerovi

Srovnávané parametry:

1. Ikonka indikující stav vozidla - Rozumí se jestli ikonka indikuje alespoň nějaký stav (stav motoru, typ jízdy, přihlášený řidič nebo definované vstupy a jiné).
2. Trasování jízdy - Jestli je k dispozici alespoň nějaké trasování (časově omezené, aktuální jízda, barevně rozlišená rychlost jízdy a jiné).
3. Přehled vozidel - Je-li k dispozici přehled všech vozidel a k nim aktuální informace.
4. Popisek u vozidla - Jsou-li k dispozici po kliknutí na ikonku vozidla, nebo u ní nějaké aktuální informace.

Firma	Ikonka	Trasování	Přehled vozidel	Popisek
GPSDozor, AutoGPS	NE	ANO (časové)	ANO (i informace)	ANO
WebDispečink, KomTest	ANO	ANO (časové)	ANO (bez informací)	NE
FleetWare	ANO	ANO (časové)	ANO (i informace)	NE
ONISystem	ANO	NE	ANO (i informace)	NE
LogisCare	ANO	ANO (celá jízda, časově rozlišená rychlost)	ANO (i informace)	NE
GXSolution	ANO	NE	ANO (i informace)	ANO

Tabulka 1: Informace poskytované dispečerovi

2.7.2 Zobrazení historie vozidla

Srovnávané parametry:

1. Je-li historie k dispozici.
2. Způsob zobrazení - podle časového filtru, podle jízd.
3. Informace v historii - Jestli jsou k jednotlivým pozicím historie nějaké informace.

Firma	Historie	Zobrazení	Informace
GPSDozor, AutoGPS	ANO	Jízda	NE
WebDispečink, Komtest	ANO	Jízda	Rychlost
FleetWare	NE	NE	NE
ONISystem	ANO	Jízda	Rychlost
LogisCare	ANO	Jízda	Rychlost
GXSolution	ANO	Časově	Podle nastavení

Tabulka 2: Zobrazení historie vozidla

2.7.3 Uživatelské nastavení

Srovnávané parametry:

1. Úprava pracovní plochy - Možnost upravit si alespoň částečně pracovní plochu (Resize panelů, přidávání modulů a jiné).
2. Zobrazení informací - Možnost nastavit si zobrazování aktuálních informací - ať už v celkovém přehledu vozidel, nebo u jednotlivých vozů.

Firma	Pracovní plocha	Informace
GPSDozor, AutoGPS	ANO (resize)	NE
WebDispečink, Komtest	ANO (resize)	NE
FleetWare	ANO (resize)	NE
ONISystem	NE	NE
LogisCare	ANO	ANO
GXSolution	ANO	ANO

Tabulka 3: Uživatelské nastavení

Na základě této analýzy konkurence byl sestaven dispečerský modul, který bude obsahovat tyto části:

- Panel, který bude obsahovat seznam všech zařízení zákazníka, určitý druh filtrace nad zařízeními.
- Panel, který bude zobrazovat vybraná zařízení dispečerem a k nim potřebné informace.
- Panel se samotnou mapou.
- Panel s nastavením dispečerského modulu a mapy.

Návrh dispečerského modulu je detailněji popsán v kapitole 5.1.

3 Architektura projektu

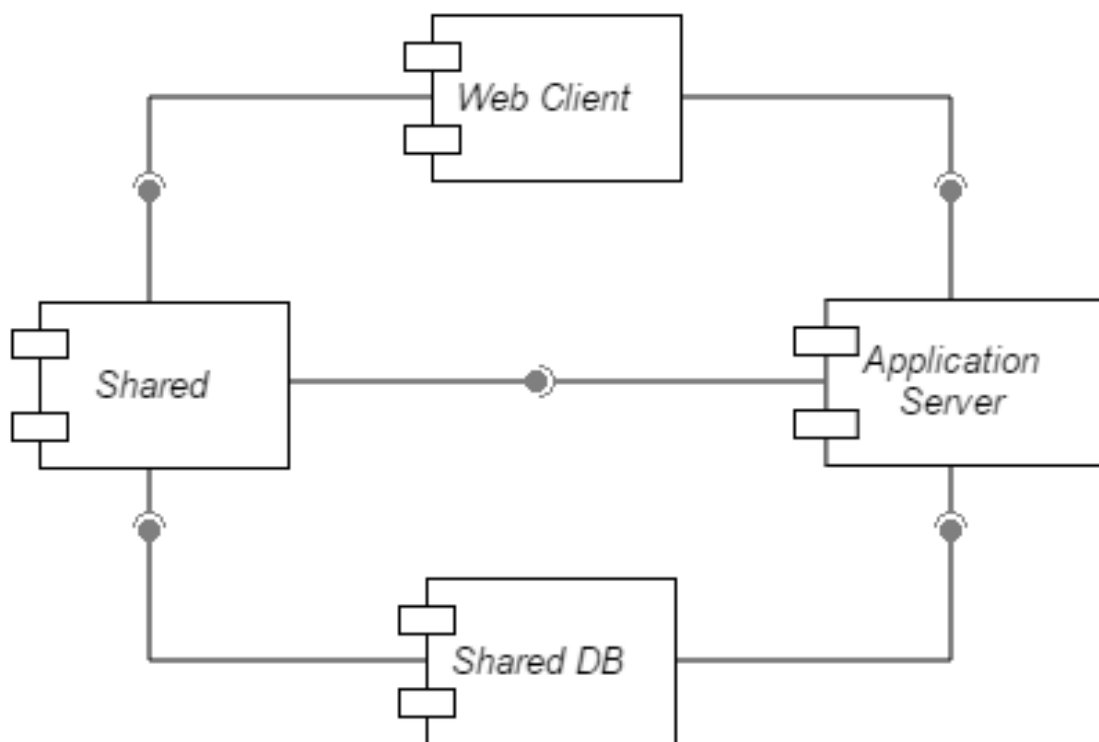
Serverová část projektu je vyvíjena v programovacím jazyce C#, je zde implementován vzor MVC. Pro přístup k databázi a komunikaci s jinými knihovnami je implementována služba WCF.

Na klientské straně jsou využity komponenty DevExpress ASP.NET MVC využívající technologii AJAX, javascriptové knihovny OpenLayers a jQuery.

Jednotlivé komponenty projektu:

- Application Server - Implementuje službu WCF.
- Shared DB - Zajišťuje komunikaci s databází pomocí odpovídajících tříd.
- Shared - Knihovna obsahující především objekty přenášející se aplikačním serverem. Dalé obsahuje nejrůznější pomocné třídy.
- Web Client - Webová aplikace, využívá technologie ASP.NET MVC.

Diagram komponent zobrazující propojení jednotlivých komponent:



Obrázek 8: Diagram komponent

4 Implementace databázové vrstvy

Kompletní, kompilovatelné zdrojové kódy tvoří neveřejnou přílohu této bakalářské práce.

Pro potřeby dispečerského modulu byla na aplikačním serveru navržena databázová vrstva, která řeší ukládání a načítání dat pro online pozice, historie pozic zařízení a uživatelského a zákaznického nastavení.

4.1 Použité technologie

Seznam použitých technologií pro implementaci databázové vrstvy:

PostgreSQL [2] je objektově-relační databázový systém s otevřeným zdrojovým kódem. Běží na všech rozšířených operačních systémech, splňuje podmínky ACID. PostgreSQL je šířen pod licencí BSD.

Entity Framework [3] je objektově-relační mapovací nástroj pro ADO.NET. Entity Framework je sada technologií pro přístup a práci k datům z databáze. Pro naše účely je implementován na databázové vrstvě a zprostředkovává veškerou interakci z databází.

4.2 Uživatelské nastavení

Uživatelské nastavení řeší ukládání a načítání dat pro nastavení aplikace při spuštění, jedná se o nastavení výchozího mapového výřezu, výchozího mapového podkladu, načtení uživatelské mapy, zobrazení mapových souřadnic a měřítka mapy.

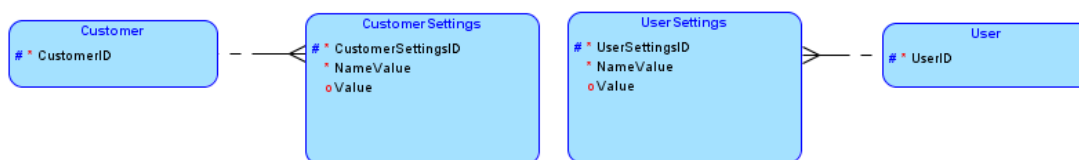
Dále se ukládají informace pro dispečerský modul, jedná se o tyto informace:

- Aktuální pozice - Barvy online a offline, velikost ikonky.
- Historie pozice - Velikost ikonky, tloušťka čáry, barvy pro jednotlivé dny a způsob barevného zobrazení.
- Uživatelská mapa - Nastavení uživatelské mapy (Název, URL, výchozí zoom, zeměpisná šířka a délka).
- Mapové výřezy - Nové mapové výřezy, editace a mazání stávajících.

Pro tyto účely jsou k dispozici dvě databázové tabulky. Tabulka pro ukládání uživatelského nastavení a tabulka pro ukládání zákaznického nastavení.

Zákaznické nastavení je společné pro všechny dispečery daného zákazníka, uživatelské je pro každého dispečera zvlášť.

Na obrázku 9 je ukázka konceptuálního modelu tabulek. U tabulek Customer a User uvádím jen primární klíče.



Obrázek 9: Konceptuální model

Na aplikačním serveru byly k těmto tabulkám vytvořeny základní funkce pro ukládání, načítání a úpravu dat.

- GetUserSettings(userID, nameValue) - Funkce vrátí uživatelské nastavení.
- GetCustomerSettings(customerID, nameValue) - Funkce vrátí zákaznické nastavení.
- GetUserSettingsList(userID, List<nameValue>) - Funkce vrátí list uživatelských nastavení.
- GetCustomerSettingsList(customerID, List<nameValue>) - Funkce vrátí list zákaznických nastavení.
- SetUserSettings(userID, nameValue, value) - Funkce nastaví uživatelské nastavení.
- SetCustomerSettings(customerID, nameValue, value) - Funkce nastaví zákaznické nastavení.
- SetUserSettingsList(userID, List<SettingsItem>) - Funkce nastaví uživatelská nastavení.
- SetCustomerSettingsList(customerID, List<SettingsItem>) - Funkce nastaví zákaznická nastavení.

4.3 Online pozice vozidel

Pro získávání dat potřebných k zobrazení online pozic zařízení byly vytvořeny dvě funkce na datové vrstvě. První funkce se stará o získání dat pro tabulku pozic. Druhá o získání dat pro ikonku, popis zařízení a bublinu s informacemi. Funkce načítají data vždy pro všechna vybraná zařízení.

4.3.1 Data pro tabulku online pozic

Tabulka online pozic zobrazuje vybraná zařízení a k nim aktuální nebo poslední uložené informace.

K získání a zpracování dat je potřeba následujících kroků:

- Zjištění přiřazených sloupců z nastavení.
- Zjištění informací pro ikonky z nastavení.
- Získání dat pro vybraná zařízení.
- Získání dat podle sloupců.

4.3.1.1 Zjištění přiřazených sloupců Dispečer má možnost nastavit si informace ze zařízení, také jakým způsobem je chce zobrazit. Každá informace je v tabulce zobrazena do jednoho sloupce. Načteme tyto informace pro dispečera, viz Výpis 1. Pokud dispečer nemá nastavené tyto informace, načteme informace nastavené zákazníkem.

```
var columns = GetRowsColumns((int)locationType, customerId, userID);
```

Výpis 1: Online pozice - Zjištění přiřazených sloupců

4.3.1.2 Zjištění informací pro ikonky V tabulce pozic se také zobrazuje stav zařízení, který je zobrazen jako ikonka. Dispečer má možnost nastavit si u ikonky barvy online a offline. Online barva se zobrazuje pokud je řidič přihlášen, offline naopak. Načtení informací, viz Výpis 2.

```
string colorOn = DispatcherModuleValues.ColorOnline;
string colorOff = DispatcherModuleValues.ColorOffline;
var settings = Database.GetUserSettingsValues(userID, new List<string>(new string[] {
    UserNames.MapColorOnline,
    UserNames.MapColorOffline
}));
foreach (var item in settings)
{
    if (item.NameValue == UserNames.MapColorOnline)
        colorOn = item.Value;
    else if (item.NameValue == UserNames.MapColorOffline)
        colorOff = item.Value;
}
```

Výpis 2: Online pozice - Zjištění nastavení pro ikonky

4.3.1.3 Získání dat pro vybraná zařízení Pro vybrané zařízení je potřeba získat data z databáze. Každé zařízení má přiřazenou jednotku, podle jednotky získáme data z databáze, viz Výpis 9.

```
List<UnitDevice> unitDevices = Database.GetSelectedActualAssignedUnitsFull(devicesID);
List<DeviceData> devices = Mapper.MapDevices(unitDevices);
```

Výpis 3: Online pozice - Získání dat pro vybraná vozidla

4.3.1.4 Získání dat podle sloupců Pro všechna vybraná zařízení je potřeba postupně načíst informace pro nadefinované sloupce, viz Výpis 4. Informace se získají z jednotek vozidel a je potřeba naformátovat jejich zobrazení.

Každý sloupec může být jinak zobrazen a to buď:

- Jenom hodnota, bez měrné jednotky (např. rychlost: 50).
- Hodnota s měrnou jednotkou (např. rychlost: 50 Km/h).
- Popis (např. Spínačka: Zapnutá).
- Barva (zobrazí se v buňce barevné pozadí podle nastavení).

```
for(int j = 0; j < devicesCount; j++)
{
    var item = devices[j];
    var unit = unitDevices.FirstOrDefault(f => f.UnitID == item.UnitID).Unit;
    for (int i = 1; i < columnsCount; i++)
    {
        if (columns[i].FieldName.StartsWith("V"))
            row.Values.Add(GetValueFromColumn(columns[i], unit, item,
                ColumnValueNum.SimpleValue));
        else if (columns[i].FieldName.StartsWith("U"))
            row.Values.Add(GetValueFromColumn(columns[i], unit, item,
                ColumnValueNum.ValueWithUnit));
        else if (columns[i].FieldName.StartsWith("D"))
            row.Values.Add(GetValueFromColumn(columns[i], unit, item,
                ColumnValueNum.Description));
        else if (columns[i].FieldName.StartsWith("C"))
            row.Values.Add(GetValueFromColumn(columns[i], unit, item,
                ColumnValueNum.ColorValue));
    }
}
```

Výpis 4: Online pozice - Získání dat podle sloupců

4.3.2 Data pro ikonku, popisek a bublinu

Ikonka indikuje stavy zařízení, online/offline stav, je v pohybu/stojí, pokud se pohybuje zobrazuje se směrová šipka. Popisek se zobrazuje u ikonky, po nakliknutí ikonky se zobrazí bublina s informacemi. Informace zobrazené v popisku a v bublině se definují v nastavení aplikace.

Načítání dat má podobnou funkcionalitu jako u online pozice a využívá se pro načtení informací popisku a bubliny. Navíc je potřeba načíst z jednotky zařízení směr pohybu a velikost ikonky, pro zobrazení pozice na mapě se musí načíst zeměpisná šířka a délka.

4.4 Historie pozic zařízení

Zobrazení historie pozic je podobné jako zobrazení online pozic, také se načítají data pro tabulku pozic a také pro ikonku a bublinu s informacemi. Historie se načítá pro jedno zařízení v časovém úseku.

4.4.1 Data pro tabulku pozic

K načtení dat je potřeba následujících kroků:

- Zjištění přiřazených sloupců z nastavení. Stejně jako u online pozic, viz kapitola 4.3.1.1.
- Zjištění informací pro ikonky z nastavení.
- Získání dat pro vybrané zařízení a přiřazené jednotky.
- Získání pozic pro vybrané zařízení a časový úsek.
- Získání dat podle sloupců.

4.4.1.1 Zjištění přiřazených sloupců

Zpracováno u online pozic, viz kapitola 4.3.1.1.

4.4.1.2 Zjištění informací pro ikonku Dispečer má možnost si v nastavení nastavit jakým způsobem se mají vybarvovat ikonky pozic. Buď podle dnů v týdnu tzn. každá pozice se vybarví barvou podle dne nebo vybarvení podle zařízení tzn. všechny pozice budou vybarveny stejnou barvou. Barvy jednotlivých dnů si samozřejmě může dispečer nastavit v nastavení. Zjištění těchto informací, viz Výpis 5.

```
string sizeWheelHistory = DispatcherModuleValues.SizeWheelHistory.ToString();
string [] daysColors;
bool paintingByDays = true;
var settings = Database.GetUserSettingsValues(userID, new List<string>(new string[] {
    UserNames.MapPaintingPositions,
    UserNames.MapDaysColors,
    UserNames.MapSizeWheelHistory
}));
for (int i = 0; i < settings.Count; i++)
{
    if (settings[i].NameValue == UserNames.MapPaintingPositions)
        paintingByDays = Convert.ToBoolean(settings[i].Value);
    else if (settings[i].NameValue == UserNames.MapSizeWheelHistory)
        sizeWheelHistory = settings[i].Value;
    else if (settings[i].NameValue == UserNames.MapDaysColors)
        daysColors = settings[i].Value.Split(';');
}
```

Výpis 5: Historie pozic - Zjištění nastavení pro ikonky

4.4.1.3 Získání dat pro vybrané zařízení K vybranému zařízení je potřeba získat data z databáze a zjištění přiřazených jednotek a následné načtení z databáze, viz Výpis 6.

```
Device deviceDB = Database.GetDevice(deviceID);
List<UnitDevice> unitDevices = Database.GetAssignedUnits(deviceID, since, until);
```

Výpis 6: Historie pozic - Získání dat pro vozidlo a jednotky

4.4.1.4 Získání pozic Podle typu jednotky a časového intervalu načteme seznam pozic z databáze, viz Výpis 7.

```
List<object> positions = new List<object>();
UnitNum unitType = (UnitNum)unitDevice.Unit.Type;
switch (unitType)
{
    case UnitNum.NotSpecified:
        break;
    case UnitNum.Teltonika_FM11:
        positions = Database.GetUnitTeltonikaPositions(unitDevice.UnitID, since, until )
            .ToList<object>();
        break;
    default:
        break;
}
```

Výpis 7: Historie pozic - Získání pozic

4.4.1.5 Získání dat podle sloupců Podobné jako u Online pozic, viz kapitola 4.3.1.4 jen se používá jiná přetížená metoda.

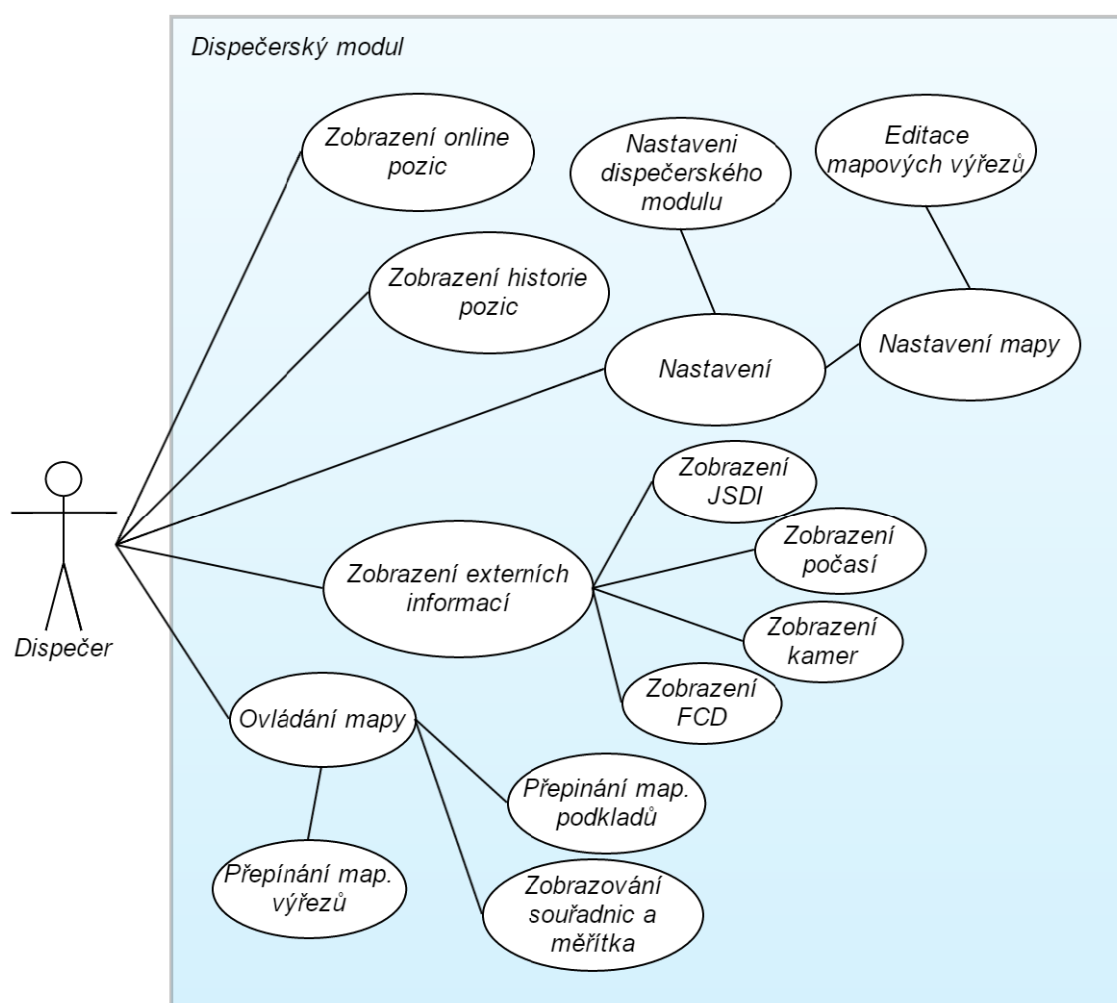
4.4.2 Data pro ikonku a bublinu

Ikonka zobrazující pozici na mapě je stejná jako v tabulce pozic, takže načítání dat je podobné jako pro tabulku pozic, jen se musí načíst další informace z nastavení, a to velikost ikonky a tloušťka čáry pro spojení pozic na mapě. Pro zobrazení pozic na mapě musíme načíst zeměpisnou šířku a délku.

5 Vytvoření uživatelského rozhraní dispečerského modulu

Uživatelské rozhraní pro dispečerský modul se věnuje především práci se zařízeními nad mapou a samotnou mapou. Dispečerský modul umožňuje dispečerovi zobrazovat online pozice vybraných zařízení, zobrazovat historii pozic zařízení, zobrazovat externí informace jako počasí, JSDI, průjezdnost silnic a kamery. Dispečerský modul také umožňuje ovládání mapy, přepínání mapových podkladů, zobrazování mapových souřadnic a měřítka mapy, přepínání mezi mapovými výřezy a vytváření nových mapových výřezů. V nastavení aplikace byla pro dispečera vytvořena záložka, která umožňuje jednoduché nastavení mapy a dispečerského modulu.

Funkcionalitu dispečerského modulu zachycuje diagram případu užití:



Obrázek 10: Dispečerský modul - Diagram případů užití

5.1 Návrh

Návrh vychází ze stávající aplikace Smart TDM společnosti GX Solutions.

5.1.1 Dispečerský modul

Pracovní plocha dispečerského modulu je rozdělena do tří částí:

- Levý panel - Slouží pro výběr vozidel, filtrování vozidel a pro výběr mapových vrstev, které zobrazují externí informace.
- Horní panel - Je tvořen záložkami. V záložkách se zobrazují tabulky online pozic, tabulky historie pozic a další záložky věnované jiným částem aplikace.
- Mapa - Zobrazuje pozice vybraných vozidel, historii pozic, externí informace a jiné informace z jiných částí aplikace.

Ovládací panel - Umožňuje jednoduché ovládání mapy, přepínání mapových výřezů a vytváření nových.

Návrh dispečerského modulu:

Mapa		Nastavení				
Skupiny ▼ Vozidla ▶ Osobní ▶ ...	Online pozice		2T2 222			
	▼	▼	▼			
Zařízení ● Škoda Octavie - 2T2 222 ● Škoda Superb - 3T3 333	<div>Mapa</div>					
Filtry 4/22/2012 4/22/2012						
Mapové vrstvy ▼ Informace ▶ Počasí ▶ ...						

Obrázek 11: Wireframe - Dispečerský modul

5.1.2 Nastavení

Mapové nastavení umožňuje nastavit si výchozí nastavení mapy, které se načte po startu aplikace.

Jedná se o:

- Mapový podklad - Výběr výchozího mapového podkladu. V nabídce jsou dva základní mapové podklady, případně uživatelský mapový podklad.
- Mapový výřez - Výběr mapového výřezu, na který se mapa vycentruje.
- Mapové souřadnice - Zapnuté nebo vypnuté.
- Měřítko mapy - Zapnuté nebo vypnuté.

Nastavení také umožňuje editaci mapových výřezů.

Náhled karty nastavení:

▼ Název	▼ Zoom	▼ Langtitude	▼ Longitude

Obrázek 12: Wireframe - Mapové nastavení

Dispečerské nastavení umožňuje dispečerovi nastavit si:

- Online pozice - Nastavení barev online a offline, velikost ikonky.
- Historie pozic - Nastavení tloušťky čáry a velikost ikonky.
- Obarvení historie - Nastavení barev pro dny v týdnu, nastavení způsobu obarvení podle dnů v týdnu nebo podle zařízení.
- Uživatelská mapa - Nastavení uživatelské mapy.

5.2 Implementace

Webový klient je vyvíjen technologií ASP.NET MVC.

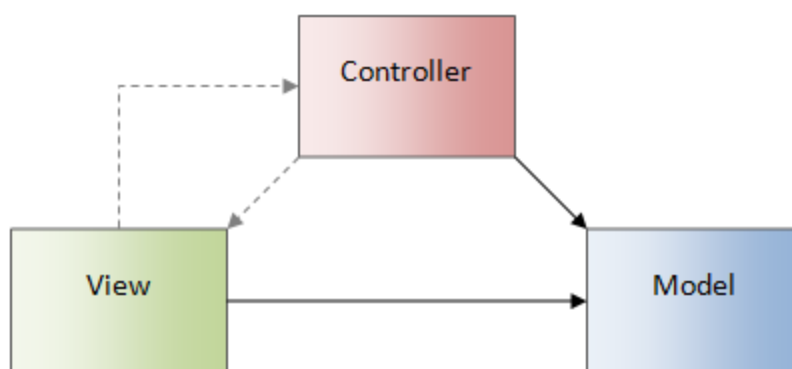
Model-View-Controller je architektonický vzor, který rozděluje aplikaci na tři logické části, které lze spravovat samostatně.

Model je část, ve které je implementována logika pracující s daty, stará se především o získávání a ukládání dat. V našem případě komunikuje především s aplikačním serverem.

View (Pohled) je část, která zobrazuje data získaná z modelu uživateli.

Controller je část, která se stará o interakci s uživatelem, zpracovává požadavky od uživatele.

Propojení jednotlivých částí:



Obrázek 13: Model-View-Controller

5.2.1 Použité technologie

Pro implementaci dispečerského modulu byly použity tyto technologie:

- JavaScript[5] - Objektově orientovaný skriptovací jazyk. Jedná se o interpretovací programovací jazyk.
- jQuery[6] - Javascriptová knihovna která usnadňuje práci s psaním kódu v Javascriptu.
- OpenLayers[7] - Javascriptová knihovna pro zobrazování a práci s mapami. V projektu je použita verze 3.
- SVG[8] - Scalable Vector Graphics, jedná se o značkovací jazyk který popisuje vektorovou grafiku. Výhodou SVG je, že se kód dá vložit přímo do HTML kódu a tím jej zobrazit.
- DevExpress ASP.NET MVC[9] - Ovládací komponenty od společnosti Devexpress. V projektu jsou použity komponenty pro technologii ASP.NET MVC.

- WMS [10] - Web Map Service - Webová mapová služba umožňuje sdílení geografické informace ve formě rastrových map v prostředí Internetu.

5.2.2 Levý panel

Součástí této bakalářské práce byla jen částečná implementace levého panelu.

Součástí práce bylo přidání položky *Historie pozic* do kontextového menu komponenty GridView, který zobrazuje všechna zařízení zákazníka a její funkcionality.

K mapovým vrstvám byly přidány položky:

- Počasí - Zobrazí na mapě aktuální počasí.
- Doprava - Zobrazí na mapě aktuální dopravní situace.
- Průjezdnost silnic - Zobrazí na mapě aktuální průjezdnost silnic.
- Kamery - Umožní zobrazit informace z kamer na křižovatkách.

5.2.2.1 Počasí Jako zdroj informací je používán server <http://openweathermap.org/>, který podporuje několik způsobů dotazování, pro naše účely je používána API pro obdélníkový výběr, to znamená, že s dotazem musíme přidat parametr *bbox*, který obsahuje souřadnice levého horního rohu a pravého dolního rohu mapového výřezu.

Data vrácená serverem jsou ve formátu JSON [4]. Přijatá data jsou rozložena na jednotlivé informace.

Pro zobrazení informací na mapě je vytvořená nová mapová vrstva pomocí třídy *ol.layer.Layer*, pro tuto vrstvu je vytvořen zdroj *ol.source.Source*.

Pro jednotlivé informace je vytvořen objekt třídy *ol.Feature*. Tomuto objektu je nastavena souřadnicemi pozice. Pro zobrazení se používá ikonka podle typu počasí. Tento objekt je poté vložen do zdroje vrstvy a zobrazen na mapě.

5.2.2.2 Doprava, průjezdnost silnic a kamery Zobrazování informací je pomocí standardu WMS [10], jako zdroj informací je používán server <http://rodosstorage.vsb.cz:8080/floreongis/>

Pro zobrazení informací stačí vytvořit novou vrstvu, jako zdroj použít třídu *ol.source.TileWMS*. Tomuto zdroji stačí nastavit url serveru, parametry a sám se postará o načítání informací.

5.2.3 Horní panel

Implementace horního panelu je pomocí devexpressové komponenty PageControl, ve které se vytvořily záložky pro online pozice a historii pozic. Do těchto záložek je implementována komponenta GridView, jedná se o dynamickou tabulku s ovládacími prvky. GridView umožňuje dynamicky zobrazit data z modelu, který je mu předán. V GridView je implementována funkce *PerformCallback*, tato funkce provádí asynchronní načítání dat pomocí technologie AJAX.

5.2.3.1 Online pozice - Nahrání dat pro tabulku pozic Při výběru zařízení v levém panelu dispečerského modulu získáme ID vybraného zařízení, toto ID se předává při volání funkce *PerformCallback* na GridView online pozic. Controller se poté postará o získání nového modelu pro GridView.

Pro GridView zobrazující online pozice využijeme třídy *DataTable.cs* jako Model, která obsahuje vlastnosti *Columns* a *Rows*, jedná se o jednoduché kolekce, které stačí naplnit daty. Data pro *Columns* získáme funkcí *GetRowsColumnsDefinitions* a data pro *Rows* získáme již předem vytvořenou funkcí, viz kapitola 4.3.1.

Ukázka funkce z Controller, který naplňuje GridView:

```
public ActionResult TablePositionPartialView(string selectedDevices)
{
    int[] devices = new int[0];
    DataTable table = null;
    if (selectedDevices != null)
    {
        devices = new JavaScriptSerializer().Deserialize<int[]>(selectedDevices);
        table = DataHelper.GetDevicesForTablePositionOnline(devices);
    }

    return PartialView("TablePositionPartial", table);
}
```

Výpis 8: Controller - PartialView pro tabulku pozic

Pro tabulku pozic bylo vytvořeno kontextové menu, které obsahuje tyto položky:

- Vlastnosti - Zobrazí stránku s vlastnostmi vybraného zařízení.
- Navigovat od vozidla - Zobrazí panel pro plánování trasy a jako počáteční bod nastaví pozici vybraného vozidla.

5.2.3.2 Online pozice - Nahrání dat pro ikonku, popis a bublinu Nahrání a zpracování dat je komplikovanější, při získání ID zařízení z levého panelu se zavolá funkce, která pomocí AJAXu předá Controlleru ID zařízení a získá zpracovaná data.

Controller využívá vytvořené funkce z kapitoly 4.3.2 pro získání dat z aplikačního serveru, tato data jsou nahraná do speciálního objektu, který slouží k serializaci do formátu JSON[4], který je poté vrácen do webového klienta.

5.2.3.2.1 Vytvoření ikonky K vytvoření ikonky se používá značkovací jazyk SVG, pomocí funkce *ConvertStatusToSVGIcon* je vygenerována ikonka podle statusu zařízení, velikost ikonky závisí od uživatelského nastavení.

Ukázka vytvoření ikonky v jazyce SVG:

```
<svg width=34 height=34 >
  <polyline points='17,17_17,0_13,5_17,0_21,5' fill='none' stroke='blue' stroke-width='2'
    transform='rotate(45,17,17)'></polyline>
  <circle cx=17 cy=17 r=11 style=' fill :rgba(51,153,102,1);' />
  <circle cx=17 cy=17 r=6.5 style=' fill :rgba(150,150,150,1);' />
</svg>
```

Výpis 9: Náhled generovaného SVG



Obrázek 14: Náhled generované ikonky

5.2.3.2.2 Zpracování získaných dat v klientovi K přidání ikonky, popisku a bubliny do mapy využijeme objektu třídy *ol.Overlay* z knihovny OpenLayers, který zobrazuje nad mapou HTML elementy, proto je potřeba vytvořit nejprve elementy, a ty následně vložit do vytvořeného objektu, který je poté přidán do mapy.

Controller nám vrací serializovaný objekt ve formátu JSON, ten je potřeba deserializovat pomocí javascriptové funkce *JSON.parse*. Informace obsažené v objektu se zpracují a vytvoří se tři HTML elementy:

- Ikonka - SVG je vložena do HTML elementu <a> pro lepší interakci.
- Popisek - Je vložen do blokového HTML elementu <div>
- Bublina - Je vložena do blokového HTML elementu <div>

Všechny tři elementy jsou vloženy do blokové elementu <div>, který obsahuje všechny vybraná zařízení. Pro každou ikonku a popisek je vytvořen objekt třídy *ol.Overlay*, kam se následně element zkopíruje. Objektu se poté nastaví souřadnice pozice, na které se má zobrazit a vloží se do mapy.

Kopírování se musí provádět. Dojde-li totiž k přepnutí záložek a je potřeba odstranit aktuální pozice z mapy - což se provádí odstraněním odpovídajících objektů - odstraní se i HTML elementy v nich obsažené. Pokud je potřeba následně zase zobrazit aktuální pozice nebude k dispozici HTML element k vytvoření nového *ol.Overlay*.

Pro bubliny je vytvořen jeden společný *ol.Overlay* objekt, do kterého se po kliknutí na ikonku zkopíruje HTML element.

Nad ikonkou bylo vytvořeno kontextové menu, které obsahuje stejné položky jako u tabulky pozic, viz kapitola 5.2.3.1.

Ukázka funkce pro vkládání vytvoření a vkládání objektu *ol.Overlay* do mapy:

```
map.AddDeviceOverlay = function(element, lon, lat)
{
    var position = ol.proj.transform([lon, lat], 'EPSG:4326', 'EPSG:3857');
    var overlay = new ol.Overlay({
        position: pos,
        element: element
    });
    map.AddOverlay(overlay);
}
```

Výpis 10: Vkládání Overlay

5.2.3.3 Historie pozic - Nahrání dat pro tabulku pozic Po zvolení položky *Historie pozic* v kontextovém menu levého panelu získáme ID vybraného zařízení. V levém panelu se nachází datumový filtr, ze kterého je nutné získat časový interval, pro který se má historie zobrazit.

Následně se zobrazí záložka pro dané zařízení a provede se *PerformCallback* na *GridView* v záložce. Controlleru se předá ID zařízení a časový interval. *GridView* používá stejný model jako *GridView* online pozice *DataTable.cs* a nahrání dat se provádí podobně - jen pro vlastnost *Rows* se využije funkce z kapitoly 4.4.1.

5.2.3.4 Historie pozic - Nahrání dat pro ikonku a bublinu Po získání ID zařízení a časového intervalu je zavolána funkce, která tyto informace předá Controlleru, který využije vytvořené funkce z kapitoly 4.4.2 pro získání dat z aplikačního serveru, data jsou vrácena ve formátu JSON[4].

5.2.3.4.1 Zpracování dat v klientovi Podobně jako u online pozic se pro zobrazení ikonky a bubliny jednotlivých pozic použije objekt třídy *ol.Overlay*. Postup přidávání ikonky je stejný jako u online pozic.

U zobrazení historie se jednotlivé pozice propojují křivkou, aby se rozpoznal jak se zařízení pohybovalo. Při zpracovávání jednotlivých pozic je nutné ukládat si souřadnice všech pozic do pole. Po zpracování všech pozic je toto pole předané funkci, která na základě souřadnic vykreslí křivku pomocí třídy *ol.interaction.Draw*, tloušťka křivky je podle uživatelského nastavení.

Ukázka funkce vykreslující křivku:

```

map.drawPositionsLine = function (data,thickness)
{
    var position = [];
    for(var i = 0; i < data.length; i++)
    {
        var pos = ol.proj.transform([data[i][0], data[i][1]], 'EPSG:4326', 'EPSG:3857');
        position.push(pos);
    }
    var f = new ol.Feature({
        geometry: new ol.geom.LineString(position, 'XY')
    });
    var line = new ol.FeatureOverlay({
        style: new ol.style.Style({
            stroke: new ol.style.Stroke({
                color: 'ff0000',
                width: thickness
            })
        }),
        features: [ f ],
        map: map
    });
    var draw = new ol.interaction.Draw({
        features: line.getFeatures(),
        type: 'LineString'
    });
    map.addInteraction(draw);
}

```

Výpis 11: Křivka historie

5.2.4 Mapa

K zobrazování mapových podkladů byla použita javascriptová knihovna OpenLayers [7] ve verzi 3. Byly použity dva základní mapové podklady:

- Navteq - Mapový podklad dostupný od společnosti GX Solutions a.s.
- OpenStreet - Volně dostupné mapové podklady.

Další mapový podklad je možné definovat v nastavení aplikace, který je společný všem dispečerům daného zákazníka.

Pro potřeby aplikace bylo upraveno zobrazování mapových souřadnic do požadovaného tvaru a zobrazení mapového měřítka.

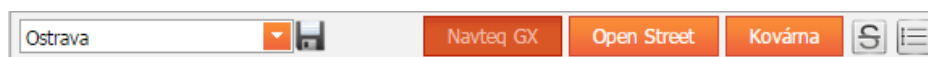
5.2.4.1 Defaultní nastavení mapy Každý Dispečer si může v nastavení aplikace nastavit defaultní nastavení mapy, které se bude zobrazovat po spuštění aplikace. Jednotlivé body nastavení byly popsány v Návrhu nastavení, viz kapitola 5.1.2.

Pro načtení nastvení byla vytvořena funkce *setUserDefaultSettings*. Funkce pomocí AJAXu odešle požadavek na Controller, který získá data z aplikačního serveru pomocí vytvořených funkcí v kapitole 4.2 data serializuje a odešle zpět na klienta.

5.2.4.2 Ovládací panel Pro jednoduché ovládání mapy byl vytvořen Ovládací panel, který obsahuje tyto ovládací prvky:

- Seznam mapových výřezů - ComboBox, který obsahuje všechny mapové výřezy dispečera, při změně volby se mapa vycentruje na zvolený výřez.
- Nový mapový výřez - Ikonka, po kliknutí na ni se zobrazí dialogové okno k vytvoření mapového výřezu.
- Tlačítka mapových podkladů - Tlačítka, která zapnou mapový podklad. Tlačítka fungují jako přepínač, aktivní může být jen jedno.
- Mapové souřadnice - Tlačítko, které zapne nebo vypne souřadnice
- Měřítko mapy - Tlačítko, které zapne nebo vypne měřítko.

Náhled ovládacího panelu:



Obrázek 15: Mapa - Ovládací panel

5.2.4.3 Vytvoření mapového výřezu Pro vytváření mapového výřezu bylo vytvořeno dialogové okno, ve kterém dispečer vyplní název mapového výřezu a zvolí, jestli se má výřez nastavit jako defaultní. Z mapy je potřeba získat střed a přiblížení aktuálního výřezu, poté jsou tyto informace pomocí AJAXu odeslány na Controller, a ten je uloží na aplikační server.

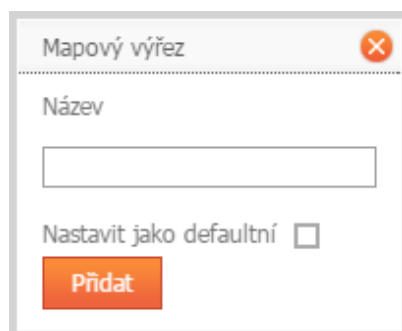
Ukázka funkce, která získá potřebné informace o výřezu:

```
function getCurrentMapCutOut() {
    var zoom = map.getView().getZoom();
    var center = ol.proj.transform(map.getView().getCenter(), 'EPSG:3857', 'EPSG:4326');

    return zoom + ";" + parseFloat(lon.toFixed(5)) + ";" + parseFloat(lat.toFixed(5));
}
```

Výpis 12: Mapa - Získání aktuálního výřezu

Ukázka dialogového okna pro vytvoření výřezu:



Mapový výřez

Název

Nastavit jako defaultní ☐

Přidat

Obrázek 16: Mapa - Mapový výřez

6 Testování vytvořené aplikace

Testování aplikace bylo prováděno v několika částech, první část byla zaměřena na funkčnost modulu a odhalování chyb, druhá část se věnovala Performance testování.

6.0.5 Testování funkčnosti modulu

Pro správnou funkčnost modulu byla do aplikace přidána testovací zařízení, ze kterých byla sbírána reálná data. Na těchto datech se například odhalily chyby ve funkcích, které se starají o načítání dat, viz Kapitola 4.

Odhalování chyb týkajících se nesprávné funkčnosti uživatelského rozhraní docházelo při práci s aplikací během implementace. Tyto chyby byly co nejrychleji odstraněny.

6.0.6 Performance testing

Performance testování se věnuje testování aplikace z hlediska výkonosti aplikace. V rámci dispečerského modulu se jednalo především o rychlost načítání dat z aplikačního serveru a následného zpracování dat v klientovi.

Pomocí refactoringu byl například změněn cyklus *foreach* na cyklus *for*, byly změněny dotazy některých funkcí, které se starají o načítání dat z databáze, omezení používání funkce *FirstOrDefault*. Všechny tyto změny měly u některých funkcí za následek až dvojnásobné zrychlení načítání dat.

Pomocí nástroje Code Analysis ve vývojovém prostředí Visual Studio byly změněny části kódu pro lepší výkon aplikace.

7 Závěr

Cílem práce bylo navrhnout a implementovat dispečerský modul pro nově vznikající systém firmy GX Solutions.

K vytvoření modulu byla potřeba důkladná analýza konkurence a srovnání se stávajícím řešením firmy GX Solutions. Na základě analýzy proběhl návrh uživatelského rozhraní a funkčnosti dispečerského modulu a následná implementace. K implementaci byly použity komponenty společnosti DevExpress a javascriptové knihovny.

Byla vytvořena databázová vrstva, ve které byly vytvořeny funkce pro načítání a ukládání dat. Vytvořené funkce se především zabývají načítáním dat pro zobrazení informací nad mapou jako online pozice a historie pozic zařízení.

Poslední část byla věnována testování a správné funkčnosti vytvořeného modulu. Během implementace bylo prováděno testování již vytvořených částí modulu, odhalené chyby byly co nejdříve opraveny. Po implementaci modulu probíhal Performance testing, který byl především zaměřen na rychlost načítání dat z databáze.

Vývoj systému probíhal v týmové spolupráci, každý člen týmu měl na starost implementaci svého modulu. U některých modulů byla pro správnou funkčnost potřeba spolupráce s jinými členy týmu.

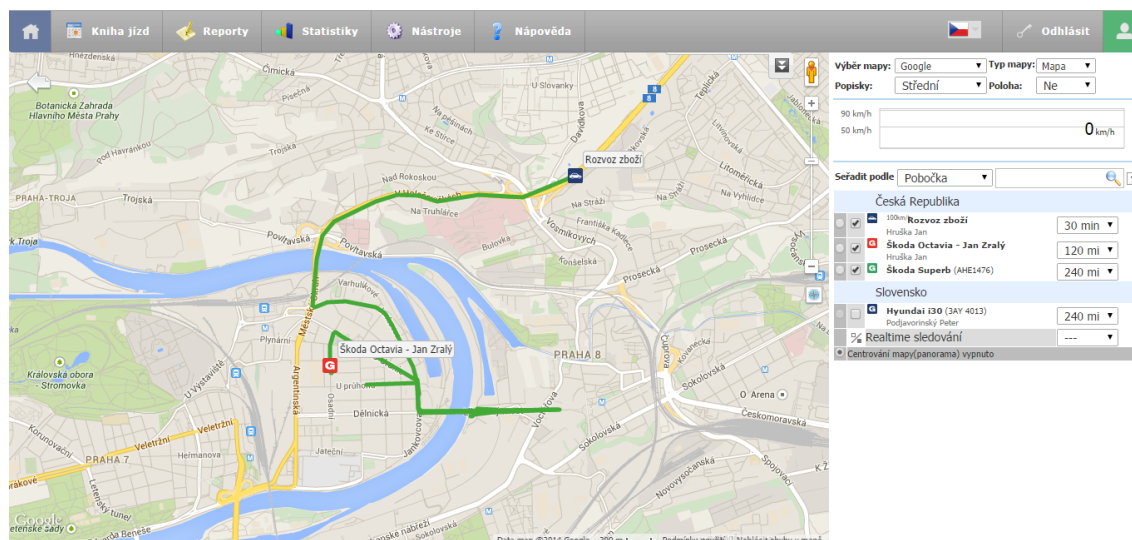
Jsem velice rád, že jsem mohl být součástí tohoto týmu a celého vývoje takto rozsáhlého systému, velice si cením zkušeností získaných při vývoji. Prozatimní vývoj byl zaměřen na vytvoření webového klienta, budoucí vývoj bude zaměřen na vývoj desktopového klienta pro specifické zákazníky.

Ondřej Vybíral

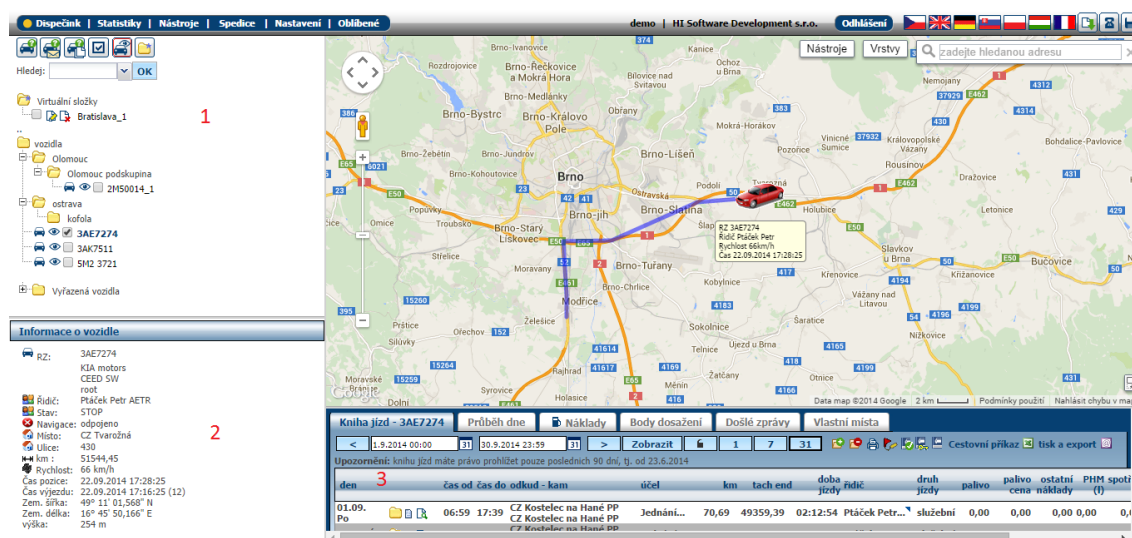
8 Reference

- [1] **JSDI:** Jednotný systém dopravních informací - [2015-04-20] Dostupné z:
<http://www.dopravniinfo.cz/jsdi>
- [2] **PostgreSQL:** Objektově-relační databázový systém - [2015-04-20] Dostupné z:
<http://www.postgresql.org/>
- [3] **Entity Framework:** Objektově-relační mapovací nástroj - [2015-04-20] Dostupné z:
<http://msdn.microsoft.com/en-us/data/ef.aspx/>
- [4] **JSON:** JavaScript Object Notation - [2015-04-20] Dostupné z:
<http://json.org/>
- [5] **JavaScript:** Skriptovací programovací jazyk - [2015-04-20] Dostupné z:
<http://www.w3schools.com/js/>
- [6] **jQuery:** Javascriptová knihovna - [2015-04-20] Dostupné z:
<https://jquery.com/>
- [7] **OpenLayers:** Javascriptová knihovna - [2015-04-20] Dostupné z:
<http://openlayers.org/>
- [8] **SVG:** Vektorový grafický formát - [2015-04-20] Dostupné z:
<http://www.w3schools.com/svg/>
- [9] **DevExpress ASP.NET MVC:** Ovládací komponenty od společnosti DevExpress - [2015-04-20] Dostupné z:
<https://www.devexpress.com/>
- [10] **WMS:** Web Map Service - [2015-04-20] Dostupné z:
<http://www.opengeospatial.org/standards/wms>
- [11] **WFS:** Web Feature Service - [2015-04-20] Dostupné z:
<http://www.opengeospatial.org/standards/wfs>

A Příloha č. 1 - Ukázka konkurenčních systémů



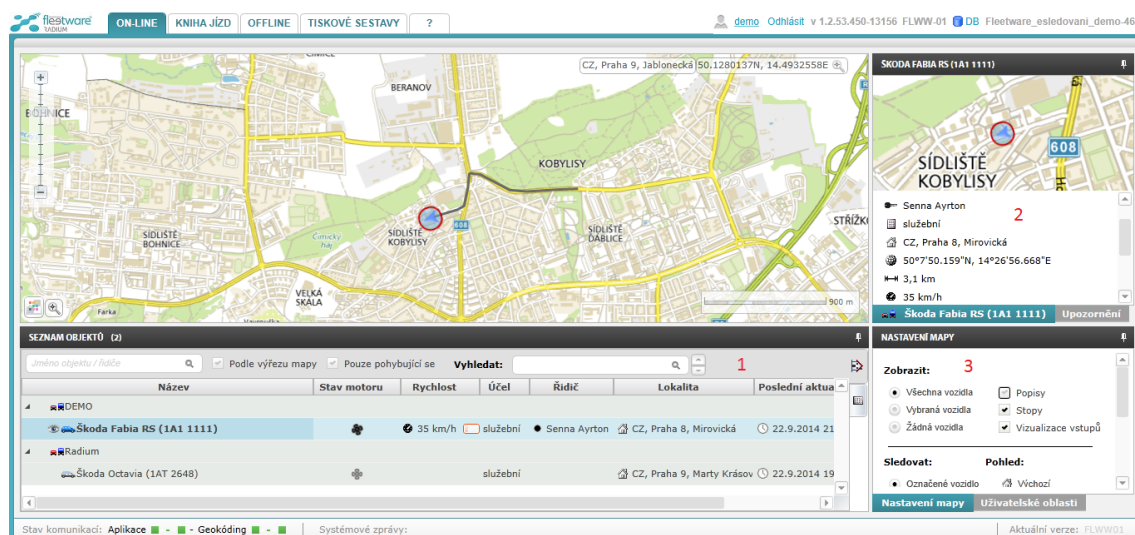
Obrázek 17: GPS Dozor - Náhled dispečerského modulu



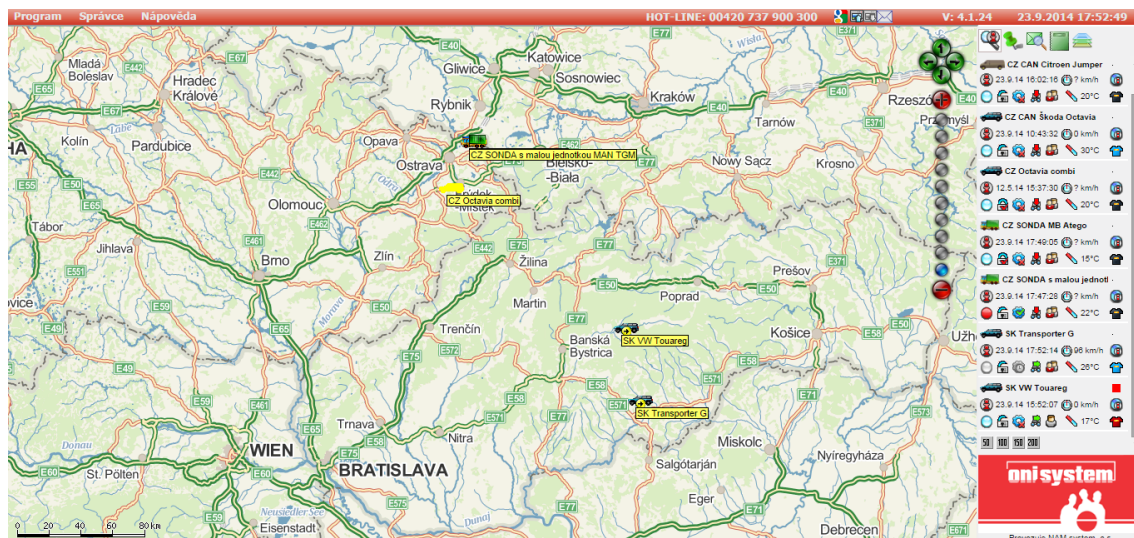
Obrázek 18: Komtest - Náhled dispečerského modulu



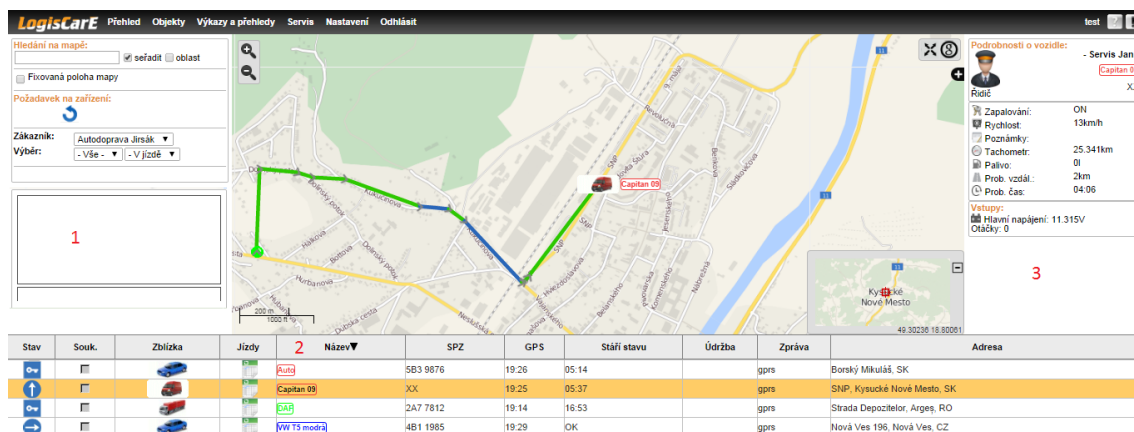
Obrázek 19: Komtest - Průběh dne



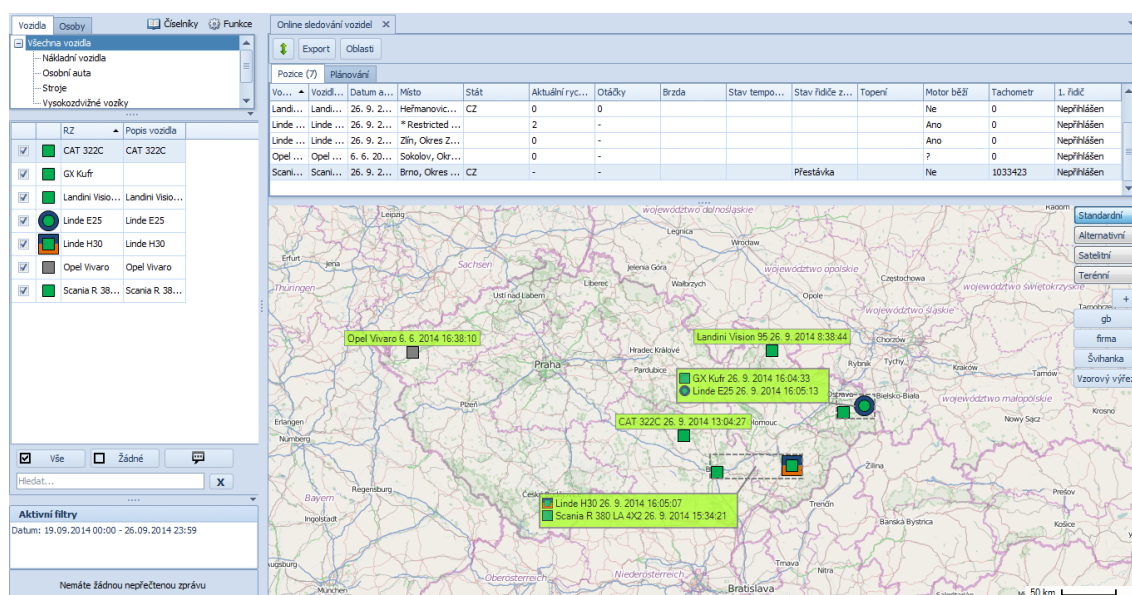
Obrázek 20: Fleetware - Náhled aplikace



Obrázek 21: ONI System - Dispečerský modul



Obrázek 22: LogisCare - Náhled aplikace



Obrázek 23: GX Solutions - Náhled aplikace